A Study of Probabilistic and Algebraic Methods for Semantic Similarity

Vasile Rus, Nobal B. Niraula and Rajendra Banjade
Department of Computer Science
The University of Memphis
Institute for Intelligent Systems
Memphis, TN 38152, USA
{vrus, nbnraula, rbanjade}@memphis.edu

Abstract

We study and propose in this article several novel solutions to the task of semantic similarity between two short texts. The proposed solutions are based on the probabilistic method of Latent Dirichlet Allocation (LDA) and on the algebraic method of Latent Semantic Analysis (LSA). Both methods, LDA and LSA, are completely automated methods used to discover latent topics or concepts from large collection of documents. We propose a novel word-to-word similarity measure based on LDA as well as several text-to-text similarity measures. We compare these measures with similar, known measures based on LSA. Experiments and results are presented on two data sets: the Microsoft Research Paraphrase corpus and the User Language Paraphrase corpus. We found that the novel word-to-word similarity measure based on LDA is extremely promising.

Introduction

We address in this paper the important task of finding the similarity of two texts using both probabilistic and algebraic methods.

Semantic similarity is an approach to the larger problem of language understanding, a core AI topic. For instance, in dialogue-based Intelligent Tutoring Systems (ITS) it is important to understand students' natural language responses. One approach to assessing students' responses is to compute how similar the responses are to benchmark solutions provided by experts (Rus & Graesser, 2006).

Below, we show an example of a real student response from an ITS and the corresponding expert-answer as authored by an expert.

Student Response: *An object that has a zero force acting on it will have zero acceleration.*

Expert Answer: *If an object moves with a constant velocity, the net force on the object is zero.*

The student response above is deemed correct as it is semantically similar to the expert answer. In general, the student response is deemed incorrect if it is not similar enough to the expert response. In this paper, we model the problem of semantic similarity as a binary decision problem in which a student response is either correct or incorrect. This type of binary modeling is similar to other semantic similarity tasks that have been proposed by the Natural Language Processing research community such as the Recognizing Textual Entailment task (Dagan, Glickman, and Magnini, 2004), the paraphrase identification task (Dolan, Quirk, & Brockett, 2004), or the student input assessment task (Rus & Graesser, 2006; McCarthy & McNamara, 2008). It should be noted that these fundamental tasks are in turn important to a myriad of real world applications such as providing evidence for the correctness of answers in Question Answering (Ibrahim, Katz, & Lin, 2003), increase diversity of generated text in Natural Language Generation (Iordanskaja, R. Kittredge, & A. Polgere, 1991), assessing the correctness of student responses in Intelligent Tutoring Systems (Graesser, Olney, Haynes, Chipman, 2005), or identifying duplicate bug reports in Software Testing (Rus et al., 2009).

The task of semantic similarity can formulated at different levels of granularity ranging from word-to-word similarity to sentence-to-sentence similarity to document-to-document similarity or a combination of these such as word-to-sentence or sentence-to-document similarity. We propose in this paper novel solutions to the task of semantic similarity both at word and sentence level. We rely on probabilistic and algebraic methods that can automatically derive meaning representations from large collection of texts in the form of latent topics or concepts.

The probabilistic method we use is Latent Dirichlet Allocation (Blei, Ng, & Jordan, 2003). LDA models documents as topic distributions and topics as distributions over words in the vocabulary. Each word has a certain contribution to each topic. Based on these distributions and contributions we define both word-to-word semantic similarity measures and text-to-text semantic similarity measures. The LDA-based word-to-word semantic similarity measure is further used in conjunction with a greedy and optimal matching method to measure similarity between larger texts such as sentences. The text-to-text measures are used by themselves to compute the similarity of texts such as between two sentences, which is our focus. The proposed semantic similarity solutions based on LDA are compared with solutions based on Latent Semantic Analysis (LSA; Landauer et al., 2007). Like LDA, LSA is fully automated. Words are represented as vectors in an LSA-derived semantic space. The dimensions of this space form latent concepts. Similarity of individual words and texts are computed based on vector algebra in this space. Given that both LDA and LSA require the specification of a desired number of latent topics or concepts a priori, an interesting question relates to which of these methods best capture the semantics of word and texts for the same number of topics or concepts. The broader question would be which of these two methods can best capture the meaning of words and texts and in what conditions. This paper is one step in that direction of elucidating the strengths and weaknesses of these methodologies for meaning inference in the context of the paraphrase identification and student input assessment tasks.

We have experimented with these methods on two different data sets: the Microsoft Research Paraphrase corpus (Dolan, Quirk, & Brockett, 2004) and the User Language Paraphrase corpus (McCarthy & McNamara, 2008). We provide experimental results on these data sets using both a greedy method and an optimal matching method based on the job assignment problem, a famous combinatorial optimization problem.

The rest of the paper is organized as in the followings. The next section provides an overview of related work. Then, we describe the semantic similarity measures based LDA and LSA. The Experiments and Results section describes our experimental setup and the results obtained. We conclude the paper with Discussion and Conclusions.

Previous Work

Given its importance, the semantic similarity problem has been addressed using various solutions that range from simple word overlap to greedy methods that rely on word-to-word similarity measures (Corley & Mihalcea, 2005) to algebraic methods (Lintean, Moldovan, Rus, & McNamara, 2010) to machine learning based solutions (Kozareva & Montoyo, 2006).

The most relevant work to ours is by Lintean et al. (2010) who looked at the role of Latent Semantic Analysis (Landauer et al., 2007) in solving the paraphrase identification task. LSA is a vectorial representation in which a word is represented as a vector in a highly dimensional space, where each dimension is believed to be representative of an abstract/latent semantic concept. Computing the similarity between two words is equivalent to computing the cosine, i.e. normalized dot product, between the corresponding word vectors. The challenge with such vectorial representations is the derivation of the semantic space, i.e. discovering the latent dimensions or concepts of the LSA space. In our work, we experimented with an LSA space computed from the TASA corpus (compiled by Touchstone Applied Science Associates), a balanced collection of representative texts from various genres (science, language arts, health, economics, social studies, business, and others).

Lintean et al. (2010) used LSA as a way to compute semantic similarity in two different ways. First, they used LSA to compute a word-to-word similarity measure which then they combined with a greedy-matching method at sentence level. For instance, each word in one sentence was greedily paired with a word in the other sentence. An average of these maximum word-to-word similarities was then assigned as the semantic similarity score of the two sentences. Second, LSA was used to directly compute the similarity of two sentences by applying the cosine (normalized dot product) of the LSA vectors of the sentences. The LSA vector of a sentence was computed using vector algebra. We will present results with these methods and with a method based on optimal matching.

LDA was rarely used for semantic similarity. To the best of our knowledge LDA has not been used so far for addressing the task of paraphrase identification, which we address here. The closest use of LDA for a semantic task was by Celikyilmaz, Hakkani-Tur, & Tur (2010) for ranking answers to a question in Question Answering (QA). Given a question, they ranked candidate answers based on how similar these answers were to the target question. That is, for each question-answer pair they generated an LDA model which then they used to compute a degree of similarity (DES) that consists of the product of two measures: sim_1 and sim_2. Sim_1 captures the word-level similarities of the topics present in an answer and the question. Sim_2 measures the similarities between the topic distributions in an answer and the question. The LDA model was generated based solely on each question and candidate answers. As opposed to our task in which we compute the similarity between sentences, the answers in Celikyilmaz, Hakkani-Tur, & Tur (2010) are longer, consisting of more than one sentence. For LDA, this particular difference is important when it comes to semantic similarity as the shorter the texts the sparser the distributions, e.g. the distribution over topics in the text, based on which the similarity is computed. Another use of LDA for computing similarity between blogs relied on a very simple measure of computing the dot product of topic vectors as opposed to a similarity of distributions (Chen et al., 2012). Similar to Celikyilmaz, Hakkani-Tur, & Tur (2010), we define several semantic similarity measures based on various distributions used in the LDA model. We do use Information Radius as Celikyilmaz, Hakkani-Tur, & Tur (2010) and, in addition, propose similarity measures based on Hellinger and Manhattan distances. Furthermore, we use LDA for measuring word-to-word similarities and use these values in a greedy and optimal matching method at sentence-level. Finally, we compare the results with LSA-based results which has not been done before to the best of our knowledge.

LDA versus LSA based Semantic Similarity

We focus in this paper on two categories of methods: those that rely on word-to-word similarity measures and those that compute similarity more globally. For instance, the word-to-word semantic similarity between two words can be computed using the cosine between corresponding vectors where each vector encodes the weights of the word along each topic or concept/dimension. Text-to-text similarity can be computed directly using the global vectors of each sentence which are obtained by summing up the individual word vectors. In LDA, global text-to-text similarity measures can be computed using the distributions over topics and over words without the need for word-to-word similarity measures.

On the other hand, the word-to-word similarity measures can be expanded to work at text-to-text level as well using greedy or optimal matching algorithms in order to compute similarity between two sentences. For instance, in a greedy method individual words in text T1 are matched to words in text T2 in a greedy way. The problem with the greedy method is that it does not guarantee a matching that provides an overall best match value for the two sentences but rather a best match for the individual words. To solve this issue we also experimented with a method that guarantees optimal overall best match using the job assignment algorithm, a well-known combinatorial optimization problem. The job assignment algorithm optimally assigns workers to jobs based on the fitness of the workers' skills to the jobs requirements (Kuhn, 1955; Dasgupta et al., 2009). In our case, we would like to optimally match words in text T1 (the workers) to words in text T2 (the jobs) based on how well the words in T1 (the workers) fit the words in T2 (the jobs). The fitness between the words is nothing else but their word-to-word similarity according to some metric of word similarity, in our case LDA or LSA-based word-to-word measures.

Greedy Matching

In the greedy approach words from one sentence (usually the shorter sentence) are greedily matched, one by one, starting from the beginning of the sentence, with the most similar word from the other sentence. In case of duplicates, the order of the words in the two sentences was important such that the first occurrence matches with the first occurrence and so on. To be consistent across all methods presented here and for fairness of comparison across these methods, we require that words must be part of at most one pair. It should be noted that others, e.g. Corley and Mihalcea (2005), did not impose such a requirement and therefore some words could be selected to be part of more than one pair.

The greedy method has the advantage, over the other methods, of being simple and fast, while also effectively using the natural order of words within the sentence, which partially encodes the syntactic information between them. The obvious drawback of the greedy method is that it does not aim for a global maximum similarity score. The optimal methods described next solve this issue.

Optimal Matching

The optimal method aims at finding the best overall word-to-word match, based only on the similarities between words. This is a well-known combinatorial optimization problem. The assignment problem is one of the fundamental combinatorial optimization problems and consists of finding a maximum weight matching in a weighted bipartite graph. Given a complete bipartite graph, *G = (S, T, E)*, with n worker vertices (*S*), n job vertices (*T*), and each edge $e_{s\in S,t\in T} \in E$ having a non-negative weight $w(s,t)$ indicating how qualified a worker is for a certain job, the task is to find a matching M from S to T with maximum weight. In case of different numbers of workers or jobs, dummy vertices could be used.

The assignment problem can be formulated as finding a permutation π for which $S_{OPT} = \sum_{i=1}^{n} w(s_i, t_{\pi(i)})$ is maximum. Such an assignment is called optimum assignment. An algorithm, the Kuhn-Munkres method (Kuhn, 1955), has been proposed that can find a solution to in polynomial time (see Dawes, 2011 for a complete formal description of the algorithm).

In our case, we modeled the semantic similarity problem as finding the optimum assignment between words in one text, T_1, and words in another text, T_2, where the fitness between words belonging in opposite texts can be measured by any word-to-word semantic similarity function. That is, we are after a permutation π for which

$S_{OPT} = \sum_{i=1}^{n} word\text{-}sim(v_i, w_{\pi(i)})$ is maximum where *word-sim* can be any word-to-word similarity measure, and *v* and *w* are words from the texts T_1 and T_2, respectively.

Latent Dirichlet Allocation

Latent Dirichlet Allocation (Blei, Ng, and Jordan, 2003) is a probabilistic topic model. Topic models aim at discovering themes in documents based on a statistical analysis of large collections of documents.

In LDA, documents are assumed to be generated based on a per document distribution over topics, θ_d. That is, each document is assumed to have its own topic distribution θ_d. Given a token or word position w_i in the document, a particular topic *t* is assigned to that position based on the per document topic distribution θ_d. The probability of topic *t* in document *d* is denoted as $\theta_d(t)$. Once the topic is chosen, a word is assigned based on the distribution over words for that topic φ_t. The probability of word w_i in topic *t* is denoted as $\varphi_t(w_i)$. LDA ignores the order of the words in a document. Also, the basic model requires the user specify a priori the number of the topics. Given that the basic LSA model works on the same assumptions, it would interesting to compare the two methods when same number of topics and concepts are specified for LDA and LSA, respectively.

LDA-based Semantic Similarity

As we already mentioned, LDA is a probabilistic generative model in which documents are viewed as distributions over a set of topics and each word in a document is generated based on a distribution over words that is specific to each topic.

A first semantic similarity measure among words would then be defined as a dot-product between the corresponding vectors representing the contributions of each word to a topic. It should be noted that the contributions of each word to the topics does not constitute a distribution, i.e. the sum of contributions does not add up to 1. Assuming the number of topics T, then a simple word-to-word measure is defined by the formula below.

$$LDA - w2w(w,v) = \sum_{t=1}^{T} \varphi_t\,(w)\varphi_t\,(v)$$

More global text-to-text similarity measures could be defined in several ways. Because a document is a distribution over topics, the similarity of two texts needs to be computed in terms of similarity of distributions. The Kullback-Leibler (KL) divergence defines a distance, or how dissimilar, two distributions *p* and *q* are as in the formula below.

$$KL(p,q) = \sum_{i=1}^{T} p_i \log \frac{p_i}{q_i}$$

If we replace *p* with θ_d (text/document *d*'s distribution over topics) and *q* with θ_c (text/document *c*'s distribution over topics) we obtain the KL distance between two documents (documents *d* and *c* in our example). Furthermore, KL can be used to compute the distance between two topics using their distributions over words (φ_{t1} and φ_{t2}). All our results reported here for LDA similarity measures based on distribution distances between two documents *c* and *d* is computed by multiplying the similarities between the distribution over topics (θ_d and θ_c) and distribution over words (φ_{t1} and φ_{t2}). For space reasons, we do not provide all the details.

The KL distance has two major problems. In case q_i is zero KL is not defined. Furthermore, KL is not symmetric which does not fit well with semantic similarity measures which in general are symmetric. That is, if text A is a paraphrase of text B that text B is a paraphrase of text A. The Information Radius measure solves these problems by considering the average of p_i and q_i as below.

$$IR(p,q) = \sum_{i=1}^{T} p_i \log \frac{2 \times p_i}{p_i + q_i} + \sum_{i=1}^{T} q_i \log \frac{2 \times q_i}{p_i + q_i}$$

The IR can be transformed into a similarity measure as in the following (Dagan, Lee, & Pereira, 1997):

$$SIM(p,q) = 10^{-\delta IR(c,d)}$$

The Hellinger distance between two distributions is another option that allows avoiding the shortcomings of the KL distance.

$$HD(p,q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{1}^{T} (\sqrt{p_i} - \sqrt{q_i})^2}$$

The Hellinger distance varies from 0 to 1 and is defined for all values of p_i and q_i. A value of 1 means the distance is maximum and thus the distributions are very different. A value of 0 means the distributions are very similar. We can transform the Hellinger distance into a similarity measure by subtracting it from 1 such that a zero distance means a large similarity score and vice versa.

Lastly, we used the Manhattan distance between distributions *p* and *q* as defined below.

$$MD(p,q) = 2 \times (1 - \sum_{1}^{T} \min(p_i, q_i))$$

MD is symmetric, defined for any values of p and q, and ranges between 0 and 2. We can divide MD by 2 and subtract from 1 to transform it into a similarity measure.

	Accuracy	Precision	Recall	F-Measure	Kappa
Baseline	66.5	66.5	100	79.9	0
BLEU	66.5	66.5	100	79.9	0
LSA	**73.5**	75.3	**89.5**	**81.8**	34.6
LSA Greedy	72.8	75.5	87.6	81.1	33.8
LSA Optimal	73.0	76.7	85.3	80.0	35.9
LDA-IR	66.5	66.5	100	79.9	0
LDA-Hellinger	66.5	66.5	100	79.9	0
LDA-Manhattan	66.5	66.5	100	79.9	0
LDA-Greedy	73.1	75.5	88.0	81.3	34.2
LDA-Optimal	73.1	**77.2**	84.4	80.7	**36.8**

Table 1. Results on the testing subset of the Microsoft Research Paraphrase corpus.

LSA-based Semantic Similarity

As we already mentioned, LSA represents the meaning of words as a vector in multi-dimensional space. The number of dimensions is an input parameter that usually is chosen to be around 300. The similarity between two words is computed as the cosine between the word's LSA vectors. The similarity between two texts is computed as the cosine between the texts' vectors which are obtained by adding the individual word vectors (Landauer et al., 2007; Lintean & Rus, 2011).

The above word-to-word and text-to-text similarity measures based on LSA and LDA have several advantages compared to, for instance, WordNet-based similarity measures (Pedersen, Patwardhan,& Michelizzi, 2004). LSA as well as LDA enable the computation of similarity measures for adjectives and adverbs too, not only nouns and verbs which is typical of many WordNet-based word-to-word similarity metrics. Another advantage of the LDA- and LSA-based metrics is that similarity measures can be computed between virtually any two words, e.g. between a noun and an adverb.

Experiments and Results

We present in this section results with the previously described methods on two datasets: the Microsoft Research Paraphrase corpus (MSRP; Dolan, Quirk, and Brockett 2004) and the User Language Paraphrase Corpus (ULPC; McCarthy & McNamara, 2008).

The MSRP corpus is the largest publicly available annotated paraphrase corpus and has been used in most of the recent studies that addressed the problem of paraphrase identification. The corpus consists of 5,801 sentence pairs collected from newswire articles, 3,900 of which were labeled as paraphrases by human annotators. The whole set is divided into a training subset (4,076 sentences of which 2,753, or 67.5%, are true paraphrases), and a test subset (1,725 pairs of which 1,147, or 66.5%, are true paraphrases). The average number of words per sentence is 17 in this corpus. A simple baseline for this corpus is the majority baseline, where all instances are classified as positive. The baseline gives an accuracy and precision of 66.5% and perfect recall.

The ULPC corpus contains pairs of target-sentence and student response texts. These pairs have been evaluated by expert human raters along 10 dimensions of paraphrase characteristics. In current experiments we evaluate the LSA scoring system with the dimension called "Semantic Completeness" for consistency purposes. This dimension measures the semantic equivalence between the target-sentence and the student response on a binary scale, similar to the scale used in MSR. From a total of 1,998 pairs, 1,436 (71%) were classified by experts as being paraphrases. The data set is divided into three subsets: training (1,012 instances, 708-304 split of TRUE-FALSE paraphrases), validation (649 instances, 454-195 split), and testing (337 instances, 228-109 split). The average number of words per sentence is 15.

All the results were obtained using 300 dimensions for LSA and a similar number of topics for LDA. This is the standard value used by LSA researchers. This number of topics has been empirically found.

We follow a training-testing methodology according to which we train first to learn some parameters of the proposed model after which we used the learned values for the parameters on testing data. In our case, we learn a threshold for the similarity above which a pair of sentences is deemed a paraphrase and any score below the threshold means the sentences are not paraphrases. We report performance of the various methods using accuracy (percent of correct predictions), precision (the percentage of correct predictions out of the predicted positives), recall (percentage of correct predictions out of all true positives), F-measure (harmonic mean of precision and recall), and kappa statistics (a measure of agreement between our method's output and experts' labels while accounting for chance agreement).

	Accuracy	Precision	Recall	F-Measure	Kappa
Baseline	67.6	67.6	100	80.7	0
BLEU	67.6	67.6	100	80.7	0
LSA	**77.7**	**77.03**	95.6	**85.3**	**41.4**
LSA Greedy	75.3	73.4	99.5	84.5	30.1
LSA Optimal	75.9	75.6	95.1	84.2	36.0
LDA-IR	67.6	67.6	100	80.7	0
LDA-Hellinger	67.6	67.6	100	80.7	0
LDA-Manhattan	67.6	67.6	100	80.7	0
LDA-Greedy	76.8	75.8	**96.4**	84.9	37.8
LDA-Optimal	75.9	75.9	94.2	84.1	36.7

Table 2. Results on the testing subset of the User Language Paraphrase corpus.

Discussion and Conclusions

From Tables 1 and 2 we can see that text-to-text similarity measures based on distribution similarities over topics and words (LDA-IR, LDA-Hellinger, LDA-Manhattan) are not very effective. This is due to the relative large number of topics (300) relative to the text sizes (i.e., sentences) which results in distributions with many very small probabilities. This number of topics was so chosen to match the dimensionality of the LSA space, as we already mentioned. We also notice from the tables that when LDA is used with the word-to-word measure in conjunction with the Greedy and Optimal matching algorithms, the results are competitive with LSA. We plan to address this problem in the future by experimenting with fewer number of topics for short texts.

Acknowledgments

This research was supported in part by Institute for Education Sciences under awards R305A100875. Any opinions, findings, and conclusions or recommendations expressed in this material are solely the authors' and do not necessarily reflect the views of the sponsoring agencies.

References

Blei, D.M., Ng, A.Y., & Jordan, M.I. 2003. Latent dirichlet allocation, The Journal of Machine Learning Research 3, 993-1022.

Celikyilmaz, A., Hakkani-Tür, D., & Tur, G. 2010. LDA Based Similarity Modeling for Question Answering, NAACL-HLT. Workshop on Semantic Search, Los Angeles, CA, June 2010.

Chen, X., Li, L., Xiao, H., Xu, G., Yang, Z., Kitsuregawa, M. (2012). Recommending Related Microblogs: A Comparison between Topic and WordNet based Approaches. Proceedings of the 26th International Conference on Artificial Intelligence.

Corley, C., and Mihalcea, R. 2005. Measuring the Semantic Similarity of Texts. In Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment. Ann Arbor, MI.

Dagan, I.; Glickman, O.; and Magnini, B. 2004. Recognizing textual entailment. In http://www.pascalnetwork.org/Challenges/RTE.

Dagan, I., Lee, L., Pereira, F.C.N. 1997. Similarity Based Methods For Word Sense Disambiguation, ACL, 1997, p. 56-63.

Dolan, B.; Quirk, C.; and Brockett, C. 2004. Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. COLING 2004, Geneva, Switzerland.

Graesser, A. C.; Olney, A.; Haynes, B. C.; and Chipman, P. 2005. Autotutor: A cognitive system that simulates a tutor that facilitates learning through mixed-initiative dialogue. In Cognitive Systems: Human Cognitive Models in Systems Design.

Ibrahim, A.; Katz, B.; and Lin, J. 2003. Extracting structural paraphrases from aligned monolingual corpora. Proceedings of the 2nd International Workshop on Paraphrasing, (ACL 2003).

Iordanskaja, L.; Kittredge, R.; and Polguere, A. 1991. Natural Language Generation in Artificial Intelligence and Computational Linguistics. Norwell, MA, USA: Kluwer Academic Publishers. chapter Lexical selection and paraphrase in a meaning-text generation model, 293–312.

Kozareva, Z. and Montoyo, A. 2006. Paraphrase Identification on the basis of Supervised Machine Learning Techniques. Proceedings of the 5th International Conference on Natural Language Processing (Fin-TAL 2006), pages 524-233.

Landauer, T.; McNamara, D. S.; Dennis, S.; and Kintsch, W. 2007. Handbook of Latent Semantic Analysis. Mahwah, Lintean, M., Moldovan, C., Rus, V., & McNamara, D. (2010). The Role of Local and Global Weighting in Assessing The Semantic Similarity Of Texts using Latent Semantic Analysis. Proceedings of the 23st International Florida Artificial Intelligence Research Society Conference. Daytona Beach, FL.

McCarthy, P.M.; and McNamara, D.S. 2008. User-Language Paraphrase Corpus Challenge https://umdrive.memphis.edu/pmmccrth/public/Paraphrase Corpus/Paraphrase site.htm. Retrieved 10/20/2012 online.

Pedersen, T.; Patwardhan, S.; and Michelizzi, J. 2004. Wordnet::similarity - measuring the relatedness of concepts. In Proceedings of Fifth Annual Meeting NAACL, 38–41.

Rus, V.; Nan, X.; Shiva, S.; and Chen, Y. 2009. Clustering of Defect Reports Using Graph Partitioning Algorithms. In Proceedings of the 20th International Conference on Software and Knowledge Engineering, July 2-4, 2009, Boston, MA.